NPS-EC-97-003

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Generating and Demodulating

by

Paul H. Moose

January 1997

Approved for Public Release; Distribution Unlimited.

Prepared for: NCCOSC RDTE Division

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M.J. Evans
Superintendent

R. Elster
Provost

This report was sponsored by NCCOSC RDTE Division.

Approved for Public Release; Distribution Unlimited.

The report was prepared by:

PAUL H. MOOSE
Associate Professor
Department of Electrical and
Computer Engineering

Reviewed by:

HERSCHEL H. LOOMIS, JR.
Chairman
Department of Electrical and
Computer Engineering

Released by:

4/8/97

DAVID W. NETZER
Associate Provost and
Dean of Research

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
January 1, 1997

3. REPORT TYPE AND DATES COVERED
Research (Technical) 1 Oct 96-1 Jan 97

4. TITLE AND SUBTITLE
Generating and Demodulating M-ary FSK-PSK Using the FFT

5. FUNDING NUMBERS
N0001497WX20022AA

6. AUTHOR(S)
Paul H. Moose

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-EC-97-003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NCCOSC RDTE Division
53560 Hull Street
San Diego, CA 93152-52001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

This paper discusses a technique for modulating and demodulating M-ary FSK using an FFT based modem typical of Coded Orthogonal Frequency Division Modulation (COFDM) systems. COFDM is one of the more promising spectrally efficient, high data rate modulation techniques for line-of-sight communications between mobile platforms. This paper will show that legacy FSK radios like the AN/MRC-142 (binary FSK modulation at 144, 288, 576 kbps) used by the U.S. Marine Corps could be easily implemented in a COFDM modem orginally designed for higher data rates and better spectral efficiency than the legacy radio. In addition, the performance of such an implementation is analyzed in detail and shown to result in negligible performance degradation (0.05 dB or less). Digital processing speed requirements are analyzed and shown to be similar to digital implementation of conventional FSK receivers. MATLAB code is included that simulates the modem.

14. SUBJECT TERMS
digital communications, modem, FFT

15. NUMBER OF PAGES
30

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

ABSTRACT

This paper discusses a technique for modulating and demodulating M-ary FSK using an FFT based modem typical of Coded Orthogonal Frequency Division Modulation (COFDM) systems. COFDM is one of the more promising spectrally efficient, high data rate modulation techniques for line-of-sight communication between mobile platforms. This paper will show that legacy FSK radios like the AN/MRC-142 (binary FSK modulation at 144, 288, 576 kbps) used by the U.S. Marine Corps could be easily implemented in a COFDM modem originally designed for higher data rates and better spectral efficiency than the legacy radio. In addition, the performance of such an implementation is analyzed in detail and shown to result in negligible performance degradation (0.05 dB or less. Digital processing speed requirements are analyzed and shown to be similar to digital implementation of conventional FSK receivers. MATLAB code is included that simulates the modem.

Generating and demodulating M-ary FSK-PSK using the FFT.

Paul H. Moose

Naval Postgraduate School

Monterey, CA.

January 10, 1997

1.Introduction

The inverse FFT can be used to generate M-ary FSK and the FFT can be used to demodulate the received signal in the following way. We begin by recognizing that an inverse DFT creates N digital carrier frequencies with amplitudes and phases determined by the frequency domain vector X of length N. M-ary FSK requires that one of $M=2^q$ carrier frequencies be transmitted for each symbol where q bits are sent per symbol. Therefore, for each symbol, X is filled with all zeros except a one is placed in the position corresponding to the frequency to be sent for the corresponding input symbol. M of the N possible frequencies will be used, but only one frequency on any given symbol. One or two additional bits/symbol can be transmitted by loading the position in the frequency array with ± 1, FSK-BPSK or ± 1 or $\pm j$, FSK-QPSK instead of simply a one. No additional bandwidth is required.

The receiver simply computes the FFT of the received symbols and extracts from this vector of length N a vector of length M of the positions corresponding to the M transmit frequencies. The FFT amounts to implementation of a correlator for each of the N frequencies. We select from those the M that could have possible been sent and proceed to decode M-ary FSK in accordance with the same principles used for any correlator (matched filter) receiver for M-ary orthogonal signaling. The M-ary FSK bits can be detected either coherently or incoherently; the PSK bits must be extracted coherently or differential coherently.

We shall refer to this type of modem as a discrete multi-tone (DMT) modem. The same modem may be used for OFDM. Diagrams of the transmitter and receiver are

shown in Fig. 1.1 and Fig. 1. 2 below.

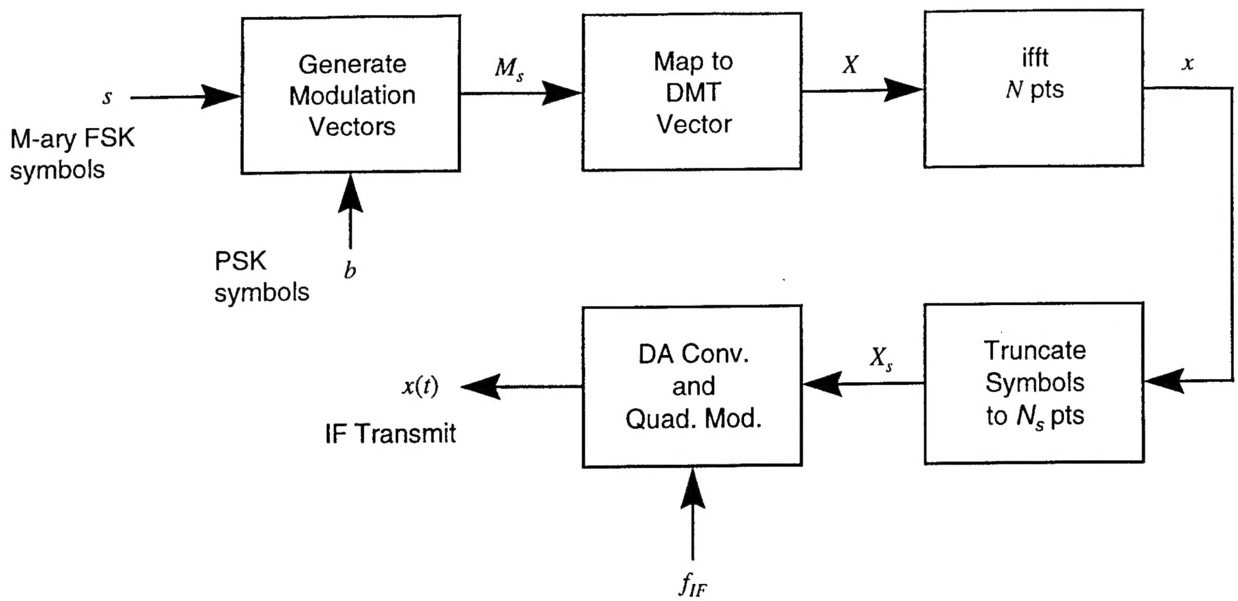

Figure 1.1 DMT Transmit

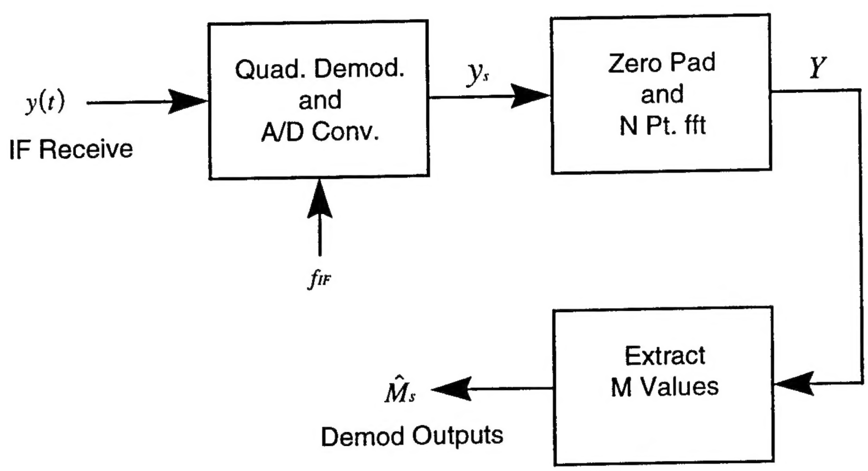

Figure 1.2 DMT Receive

2. Frequency spacing and symbol rate

Let the symbol rate be r_b and the separation of the frequencies in the M-ary FSK be del_f. Refer to Fig. 2.1 below. Now define

$$P=\text{ceil}(cyc) \tag{2.1}$$

where cyc = del_f/r_b and ceil means next largest integer. P gives the number of cycles of del_f in 1 symbol interval to the next whole number. Now let

$$\alpha= P\text{-}cyc \tag{2.2}$$

be the fraction of the last whole cycle that must be removed to achieve the correct symbol interval length $T_b=1/r_b$. Let us assume we will use an N point FFT to generate the required frequency as described above. Let

$$T=N^*\Delta t = P^*T_o \tag{2.3}$$

where $\Delta t = 1/f_s$, is the sampling interval (the reciprocal of the sampling frequency) for the FFT output and T_o=1/del_f is the reciprocal of the frequency spacing. Next let

$$N_s = \text{round}(T_b/\Delta t) \tag{2.4}$$

determine the number of discrete time points in a symbol to the nearest integer value. We shall transmit only the first N_s points of the N points output from the inverse FFT. However, due to the rounding in (2.4), the symbol interval can be in error by as much as half a sampling interval. We see that it is impossible in general to obtain exact values of symbol rate and frequency spacing simultaneously due to the fact the symbols are being generated in discrete time.

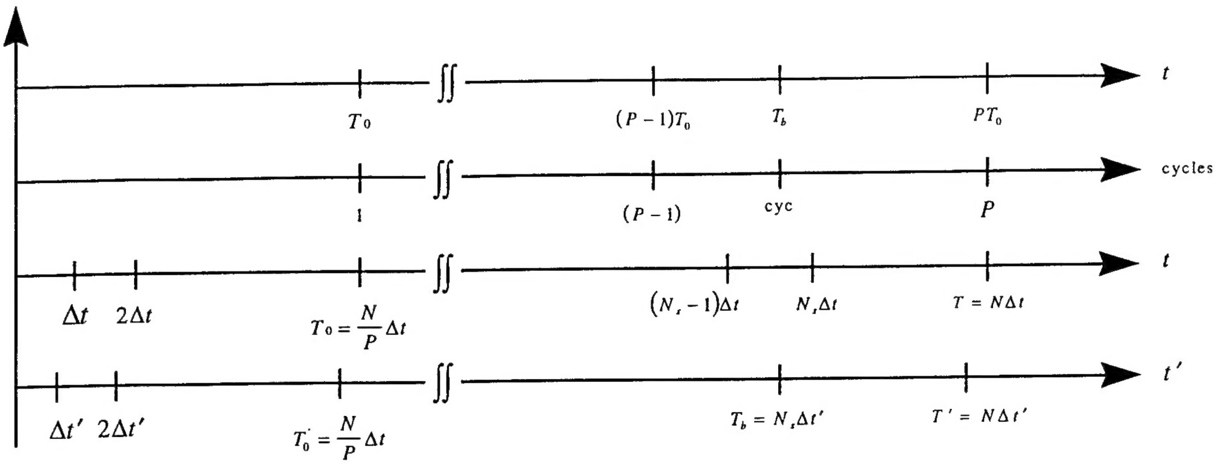

Fig. 2.1 Timing relationships

We may choose to make the symbol rate exact or the frequency spacing exact and accept the resulting error in the other. Assume we elect to make the symbol rate exact. We accomplish this by changing the sampling interval from Δt to Δt' such that

$$N_s = T_b/\Delta t', \tag{2.5}$$

which alters the sampling frequency to

$$f_s' = N_s * r_b \tag{2.6}$$

and the frequency separation to

$$del\_f' = f_s'*P/N. \tag{2.7}$$

Alternatively, we can keep the original design frequency separation del_f and accept the symbol interval which is slightly in error. There are two ways we might want to do this. First, we can modify the roundoff of (2.4) to

$$N_s = \text{floor}(T_b/\Delta t) \tag{2.8}$$

which rounds toward zero such that the new symbol interval

$$T_b' = N_s * \Delta t \tag{2.9}$$

is always equal to or less than the design interval and the new symbol rate

$$r_b' = f_s/N_s \tag{2.10}$$

is greater than or equal to the design symbol rate. The transmission can be augmented with stuff symbols to synchronize the two rates. This digital technique may be desirable in the event we wish to accommodate plesiosynchronous sources.

The second approach is to roundoff to the next greater integer value by modifying (2.4) to

$$N_s = \text{ceil}(T_b/\Delta t) \tag{2.11}$$

such that the symbol interval is slightly greater than or equal to the design interval and then truncate the symbols in the analog domain, after the D/A converters, to the correct length using the actual symbol rate clock.

3. Frequency separation error

The discrete time errors induced by the roundoff procedures suggested above are related to the sampling interval used which in turn is related to the length of the FFT used. We can bound the frequency separation error

$$\epsilon = (\text{del\_f} - \text{del\_f'})/\text{del\_f} \qquad (3.1)$$

as follows. Since del_f'/del_f = $\Delta t/\Delta t'$, and $\Delta t' = T_b/N_s$, then del_f'/del_f = $N_s*\Delta t/T_b$. But $| T_b - N_s*\Delta t | \leq ½(\Delta t)$ so that

$$|\epsilon| \leq ½(\Delta t)/T_b. \qquad (3.2)$$

Now $T = \Delta t*N$ and $T = P*T_o$ so that $\Delta t = P*T_o/N$. Substituting into (2.12) and using (2.1) gives

$$|\epsilon| \leq 1/2\{\text{ceil(cyc)/cyc}\}/N. \qquad (3.3)$$

We see that the maximum error is, as expected, inversely proportional to N the number of points in the FFT. The maximum error and actual errors obtained for length FFTs from 16 to 256 and for three different symbol rates of 144, 288 and 576 Kbps and a fixed frequency separation of 400 Khz are listed in Table 3.1 below.

4.0 Effects of the frequency separation error

The effects of the frequency separation error on the performance of the M-ary FSK transmission system may be analyzed as follows. If a DMT transmitter is communicating with a DMT receiver, there is no loss of performance as both transmitter and receiver are using the same frequency separation. The only effect is a minor change of the occupied bandwidth. However, if a conventional M-ary FSK modem is transmitting with a frequency separation del_f and the DMT is receiving with a frequency separation del_f' or if a DMT transmitter is sending with a frequency separation del_f' and an conventional modem

N Pts in FFT	Symbol Rate Kbps	Max Error (ε del_f) Khz	Actual Error (ε del_f) Khz
8	144	27.0	22.0
8	288	36.0	32.0
8	576	36.0	32.0
16	144	13.5	5.0
16	288	18.0	4.0
16	576	18.0	4.0
32	144	6.75	5.0
32	288	9.0	4.0
32	576	9.0	4.0
64	144	3.375	1.75
64	288	4.5	4.0
64	576	4.5	4.0
128	144	1.6875	1.625
128	288	2.25	.500
128	576	2.25	.500
256	144	.84375	.0625
256	288	1.125	.500
256	576	1.125	.500

TABLE 3.1 Errors in Frequency Separation vs. N

is receiving with a separation del_f, there will be a loss in performance related to the amount of error. The cases are identical. For the purpose of analysis, consider the later

and assume the receiver is incoherent, a bank of M quadrature correlators (or equivalent) as shown in Fig. 4.1 below. Now assume that the DMT transmitter transmits the m^{th} discrete frequency at frequency f_o+m*del_f' so that the receive signal during the symbol interval in question is

$$r(t) = (2^*E_b/T_b)^{1/2} [\cos(2\pi(f_o+m^*del\_f')t +\theta] + n(t) . \qquad (4.1)$$

Consider the outputs of the m^{th} correlator. Sampled at the end of the symbol interval they will be

$$x_m = (2^*E_b/T_b)^{1/2} [\sin(2\pi\ m \in cyc-\theta) + \sin (\theta)]/[2\pi\ m \in cyc] + n_{mx} \qquad (4.2)$$

and

$$y_m = (2^*E_b/T_b)^{1/2} [\cos(2\pi\ m \in cyc -\theta) -\cos(\theta)]/[2\pi\ m \in cyc] + n_{my} \qquad (4.3)$$

where if n(t) is AWGN with PSD $N_o/2$, then n_{mx} and n_{my} are statistically independent . guassian random variables with zero means and variances N_o/T_b.

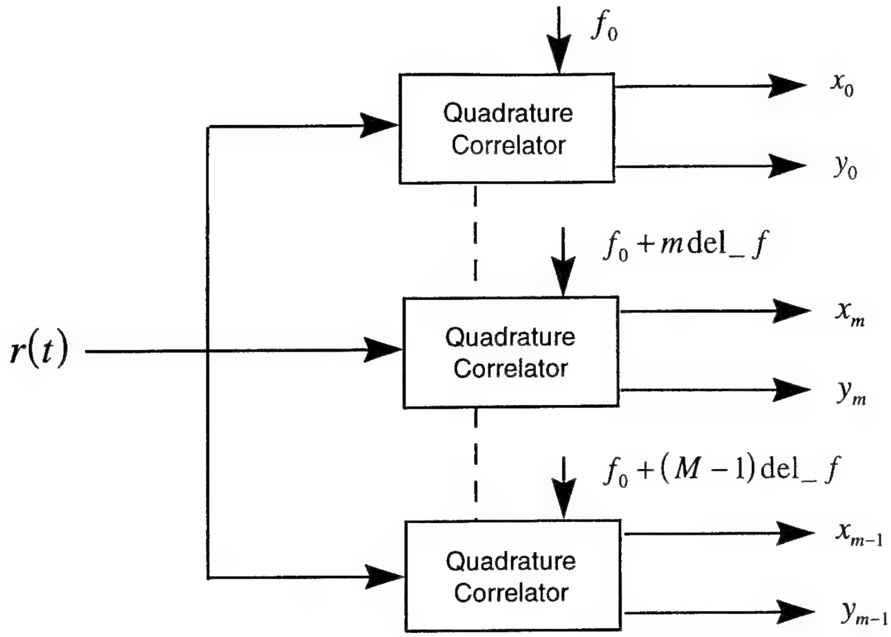

Fig. 4.1 Receiver of M Quadrature Correlators

Note that as $\epsilon \to 0$,

$$x_m \to (2*E_b/T_b)^{1/2} \cos(\theta) + n_{mx} \quad \text{and} \quad y_m \to (2*E_b/T_b)^{1/2} \sin(\theta) + n_{my}\,.$$

the theoretical outputs of the quadrature correlator with no error in the system.

Let $z_m = x_m + j\,y_m = s_m + n_m$, where s_m and n_m are the complex representations for the signal and noise components at the correlator output. An incoherent receiver processes the magnitude of z_m for decision making; a coherent receiver utilizes the magnitude and phase. The magnitude squared of the signal component is

$$|s_m|^2 = (2*E_b/T_b)[\,2\{1 - \cos(2\pi\, m\, \epsilon\, cyc)\}/\{2\pi\, m\, \epsilon\, cyc\}^2\,] \qquad (4.4)$$

which is independent of the phase θ. Eq. (4.4) may be approximated for small ϵ by

$$|s_m|^2 \approx (2*E_b/T_b)[1 - (2\pi\, m\, \epsilon\, cyc)^2/12]\,. \qquad (4.5)$$

The loss in correlator output due to frequency error from (4.4) is

$$L_{dB} = 10\log_{10}[\ 2\{1 - \cos(2\pi\ \alpha)\}/\{2\pi\ \alpha\}^2\ ] \tag{4.6}$$

with $\alpha = m\ \epsilon$ cyc. The BFSK loss (m =1) for all values shown in Table 3.1 is less than .34 dB, less than 0.13 dB for values of N of 16 or greater and less than .05 dB for values of N of 32 or greater.

5.0 Digital frequency generation and reception

Let

$$\underline{s} = [\ s0\ s1\ s2\ .\ .\ .\ .\ .\ .\ .\ .\ ] \tag{5.1}$$

be a stream of M-ary symbols to be transmitted using M-ary FSK, each symbol contained in the integers

$$sj \in \{\ 0\ ,\ P\ ,\ 2P\ ,\ .\ .\ .\ .\ ,(M-1)P'\ \}.$$

Refer to (2.1) for definition of P. We first map this stream of symbols to a stream of modulation vectors of length M. That is

$$\underline{\mathbf{Ms}} = [\ Ms0\ Ms1\ Ms2\ .\ .\ .\ .\ .\ ] \tag{5.2}$$

is a stream of column vectors of length M with a one in the $(mP)^{th}$ position if the symbol is mP and zeros in all the others. That is the j^{th} modulation vector is

$$Msj = [\ 0\ 0\ .\ .\ .\ \ 0\ 0\ 1\ 0\ .\ .\ .\ \ 0\ 0\ ]^t \tag{5.3}$$

when $sj = mP$ where the 1 is in the $(mP)^{th}$ position of (5.3).

It is possible to send v additional bits in each symbol using phase modulation by making the value in (5.3)

$$Msj_{mP} = \exp(j\phi_{mP}) \tag{5.4}$$

with

$$\phi_{mP} \in \{ 0, \Delta\phi, 2\Delta\phi, \ldots (2^{v}-1)\Delta\phi\} \tag{5.5}$$

and

$$\Delta\phi = 2\pi/2^{v} \tag{5.6}$$

instead of simply a one. For example, for BPSK (p=1) and QPSK (p=2),

$$\phi_{mP} \in \{ 0, \pi\} \text{ and } \{ 0, \pi/2, \pi, 3\pi/2 \} \tag{5.7}$$

respectively, and

$$Msj_{mP} \in \{ 1, -1 \} \text{ and } \{1, j, -1, -j\} \tag{5.8}$$

respectively.

In either event, the stream of modulation column vectors are now mapped to a stream of discrete frequency column vectors

$$\underline{\mathbf{X}} = [\, X0\ X1\ X2 \ldots\ldots\ldots] \tag{5.9}$$

where Xj is a vector of length N with all values but one equal to zero.

Recall that the baseband analog frequencies that correspond to the digital frequencies are

$$f_k \in \{0, \delta f', 2\delta f', 3\delta f', \ldots \quad (N-1)\,\delta f'\} \tag{5.10}$$

The frequencies are spaced at $1/N^{th}$ the sampling frequency so using (2.7) we see that

$$\delta f' = f_s'/N = del\_f'/P. \tag{5.11}$$

Therefore by placing the modulation values Msj, which can only be non-zero in one of every P^{th} position, into the first PM rows of Xj we will generate a single discrete multi-tone by executing the iFFT of Xj. Successive symbols will have tones separated by multiples of del_f' in accordance with the symbol stream $\underline{s}$. The modulation process is completed by transmitting only the first N_s values of the N discrete time values generated by the iFFT in order to create the correct symbol rate r_b.

Reception and de-modulation of symbols proceeds essentially in the reverse of their generation. The N_s discrete time complex baseband samples for each symbol are augmented with zeros to form vectors of length N. The FFT is computed and the values from the M positions that correspond to the M possible discrete multi-tones are extracted from each symbol and are decoded in accordance with standard procedures for M-ary signaling. Phase modulation, if any, is decoded conditionally, that is after the decision is made about which frequency has been sent.

6.0 Processing requirements

If the discrete M-ary FSK-QPSK symbols are generated and demodulated using a

conventional FFT complex radix two algorithm, then the processor must complete

$$Q = (N/2)\log_2(N) \tag{6.1}$$

complex multiplies for each symbol. Table 6.1 shows the values of Q in Mflops for different values of N and for symbol rates of 144, 288 and 576 Ksps. Note that these are the same speeds required of M-ary FSK-QPSK for any value of M <N, so BFSK is really the least efficient use of the processing power. However, if processing can be provided with standard chip sets, then efficiency is not an issue, and great flexibility is gained by utilizing the FFT technique of generation/reception.

N No. of Points in FFT	**Q Mflops (144.Ksps)**	**Q Mflops (288 Ksps)**	**Q Mflops (576 Ksps)**
8	1.728	3.456	6.912
16	4.608	9.216	18.432
32	11.520	23.040	46.080
64	27.648	55.296	110.592
128	64.512	129.024	258.048
256	147.456	294.912	589.824

Table 6.1 Processing Requirements

Processing using the FFT for M-ary FSK is inefficient for modulation but not particularly for de-modulation, as we will now demonstrate. The basic iDFT algorithm can be written in matrix form as

$$\mathbf{x} = (1/N)\, \mathbf{X}\, \mathbf{W} \tag{6.2}$$

where **X** is the 1XN row vector of modulation values, **W** is the NXN matrix of "twiddle factors", and **x** is the 1XN row vector of discrete time domain samples for a transmission symbol. The rows of **W** are the 1XN row vectors

$$\mathbf{w}_k = [\ 1\ w_{1k}\ w_{2k} \ldots\ldots w_{nk} \ldots\ldots w_{(N-1)\,k}\ ] \quad ; k = 0,1,2,\ldots N-1 \qquad (6.3)$$

where

$$w_{nk} = \exp(2\pi jnk/N). \qquad (6.4)$$

For arbitrary **X**, computation of (6.2) requires N^2 complex multiplications. The efficiency of the FFT algorithm arises from the ability to factor **W** into l sparse NXN matrices when $N = 2^l$ (the radix two algorithm) each of which require only N/2 complex multiplications (Brighham, Chapter 11.) However, in the case of M-ary FSK-QPSK, **X** contains just one non-zero element M_k in the k^{th} position which is one for M-ary FSK, ± 1 for M-ary FSK-BPSK, and ± 1 or $\pm j$ for M-ary FSK-QPSK. Thus

$$\mathbf{x} = (1/N)\, M_k\, \mathbf{w}_k \qquad (6.5)$$

which requires no complex multiplications since $\pm j\, w_{nk} = w_{nk \pm N/4}$ are also among the basic N twiddle factors that appear in each of the rows of **W**. Although all the twiddle factors appear in each row of **W**, they appear in different orders. So execution of (6.5) requires no multiplications (we ignore the 1/N gain factor), only addressing the N basic twiddle factors in correct order, similar to what is required in executing the successive stages of the FFT algorithm. Thus, the only speed limitation is memory access time for generation of M-ary FSK-QPSK.

Now consider reception of M-ary FSK-QPSK. After reduction to complex

baseband discrete time samples, we require computation of

$$\mathbf{Y} = \mathbf{y}\,\mathbf{W}^* \tag{6.6}$$

which again requires N^2 complex multiplications if computed directly or $(N/2)\log_2(N)$ complex multiplications if computed using the FFT. However, in M-ary FSK, we only are required to compute M of the N values in **Y**, that is we must implement M correlators. This requires MN complex multiplications if computed directly. Therefore for

$$M > (\tfrac{1}{2})\log_2(N) \tag{6.7}$$

direct computation is less efficient than FFT computation. Since $N = 2^l$ for l stage FFT, then direct computation is less efficient than FFT for

$$M > l/2 \tag{6.8}$$

For example, for BFSK (M=2) direct computation is less efficient than FFT computation for $l < 4$. They are equal at $l = 4$ (N=16). However, if we decide to transmit 8-ary FSK, than direct computation is less efficient than the FFT for all values of l up to 16 (N = 8192).

7.0 Conclusions

Although the FFT may be represent extremely inefficient processing for implementation of the FSK waveform generation, it has superior efficiency to a bank of M digital correlators as a receiver for all values of M greater than two and is comparable for M = 2, i.e. BFSK.

The discrete time generation of M-ary FSK, which is inherent in this technique, renders it impossible to simultaneously achieve exact symbol rates and frequency spacing

between the analog tones. However, the errors induced, which are inversely proportional to the length of the FFT used, are relatively small even for N=8, and the resulting loss in performance is less than 0.34 dB.

The primary advantage of this technique is its flexibility. With simple program control changes, the same modem can be used for M-ary FSK-PSK, and OFDM. BFSK is the simplest case of M-ary FSK.

8.0 References

Brighham, E. Oran, **The Fast Fourier Transform**, Prentice-Hall, 1976.

APPENDIX I

```
%                    DMT Modulator
%                    Written by: Paul H. Moose
%                    Naval Postgraduate School
%                    Monterey, CA
%
%       This m-file generates M-ary FSK or OFDM transmit waveforms.
%It is implemented using the inverse fft to digitally create the carrier
%frequencies required for each symbol.
%
%INPUTS:
%       type ='mfsk' or 'ofdm' to specify the desired modulation
%       q1 = no. of bits carried by each of the M1-ary input characters
%       q2 = no. of bits carried by each of the M2-ary input characters
%       KK = no. of carriers used in the OFDM symbols. Set to 0 for
%               M-FSK.
%       N = no. of points used in the ifft. Must be greater than KK for
%               OFDM. Should be at least 4*M (M=2^q) for M-FSK.
%       rb = symbol rate for M-FSK; number of points in guard interval
%               for OFDM.
%       del_f = frequency separation of carriers.
%
%       S = matrix of M-ary input characters in decimal integer
%          notation. The first row contains the M1-ary characters that will
%          be used for the FSK or OFDM. If there is a second row,
%       it contains the M2-ary characters for phase modulation of FSK-PSK.
%
%OUTPUTS:
%       x = complex baseband output sample sequence.
%       X = frequency domain array of modulation values. The columns are
%               length N and each column represents a transmission symbol.
%               x is formed from ifft(X).
%       MM = frequency domain array of modulation values used to form X.
%               For M-FSK, the columns of MM are of length M. For OFDM
%               the columns of MM are of length KK. In the case of M-FSK
%               only one of the rows of each column is non-zero, corresponding
%               to the frequency to be transmitted for that symbol. In
%               the case of OFDM, all of the rows of each column contain
%               modulation values to be transmitted. In the case of
%               M-FSK, q bits are transmitted with each symbol, while
%               in OFDM, Q*KK bits are transmitted with each symbol.
%       MP = the real output obtained from quadrature modulation of x
```

```
%               onto a carrier frequency fo (fo is currently set to
%               1200 in the program). (MP is automatically plotted if
%               there are fewer than 20 output symbols). For M-FSK only.
%
%SUBROUTINES REQD. :
%       freqa.m
%
% USEAGE:
%       [x,X,MM,MP]=dmtmod(type,q1,q2,KK,N,rb,del_f,S)
%=============================================================
========

function [x,X,MM,MP]=dmtmod(type,q1,q2,KK,N,rb,del_f,S)

%=============================================================
=========

%Initialize

[aa,cc]=size(S)

%=============================================================
=========
%=============================================================
=========

%M-FSK: Determine number of cycles of fundamental carrier and number of
%sample points to be used to account for the fractional cycle when the del_f
%is not an exact multiple of rb.


if type=='mfsk'
        if rem(del_f,rb)==0 %Use this if del_f is a multiple of rb
                P=del_f/rb;
                Ns=N;
        else % Use this when del_f is not a multiple of rb.
                cycles=del_f/rb;
                P=fix(cycles)+1;
                fraction=P-cycles;
                Ns=N-round(fraction*N/P);
                Nss=N-fix(fraction*N/P);
        end
%=============================================================
===============
```

```
% Display the Sampling frequency , Exact DEL_F,),
% Carson's Rule BW, and Bitrate.
	rbb=N*del_f/(Nss*P);
	fs=Ns*rb
	DEL_F=fs*P/N
	%BW=rb+(2^q1-1)*DEL_F
	Ns
	Bitrate=q1*rb
	rbb
%===========================================================
================

%Generate array of ones properly spaced to give next integer number of cycles
% per symbol above the correct number of cycles.


	s=S(1,:);
	s=P*s+1;



%Now compute the phase modulation, if any using the second row of S

	if aa==2

		M2=2^q2;
		delphi=2*pi/M2;
		phi=delphi*(S(2,:));
		MP=exp(2*pi*j*phi);

	else
		MP=ones(1,cc);
	end

	for n=1:cc
		MM(s(n),n)=MP(n);
	end

%===========================================================
========
% Locate carriers in the frequency domain array of N digital
% frequencies with carrier number one
%at frequency -M/2, carrier M/2 at frequency -1,carrier M/2+1
% at zero frequency and carrier M at frequency M/2-1.
```

```
% Minimum value for N is 2*M.

        X=freqa(N,MM);

%=====================================================================
========
%Take ifft of frequency domain array producing time domain array
%of cc symbols of N points each of which are one of M complex sinusoids.

        x=ifft(X);
        x=x(1:Ns,:);%Shorten the array to Ns points to remove the
                        %fractional cycle when del_f is not multiple
                        %of rb.
%=====================================================================
========
%=====================================================================
========
elseif type=='ofdm'
%=====================================================================
==========
        fs=N*del_f
        BW=KK*del_f
        Bitrate=KK*q1*del_f
%=====================================================================
===========
        %The serial symbol stream s is first inverse muxed into KK streams
        %that will be the rows of the matrix S. The columns of S
        %will be ofdm symbols.

        r=rem(cc,KK);
                if r~=0
                        disp('      ')
                        disp('Input being truncated by')
                        disp(r)
                        disp('symbols')
                end
        sof=S(1,1:cc-r);
        L=length(sof)/KK;
        Sof=reshape(sof,KK,L);
        [KK,L]=size(Sof);
%=====================================================================
=========

%Modulation values MM with amplitude one and one of 2^q=M equal
```

```
% phase values are differentially coded for each of the KK
% carriers. The first column (symbol)  is one zero phase.
% The next columns are differentially coded in phase.


        dph=2*pi/2^q1;

        SD=cumsum(Sof)'; % Differentially code the phase values

        MM= exp(i*dph*SD);% Generate the modulation values.

        MM=[ones(KK,1) MM];% Add the reference modulation values.(Should
                          %change these from all ones in the future)

%==============================================================
=========

% Locate the modulation values in the frequency domain array
% of digital carriers

        X=freqa(N,MM);

%==============================================================
===========
% Create the multiple ofdm carriers by executing the ifft and add the guard
% interval.

        x=ifft(X);

                if rb==0
                x=x;
                else
                x=[x(N-rb+1:N,:); x];
                end
end
%==============================================================
==============

% Quadrature modulate the baseband symbols onto an IF carrier frequency fo and
%if the modulation type is 'mfsk'

if type == 'mfsk'
[rr,cc]=size(x);
```

```
fo=1200
ko=fo*N/fs;
ko=floor(ko)  % The digital carrier frequency corresponding to fo
MP=x(:).';
nn=0:length(MP)-1;
MP=MP.*exp(2*pi*i*nn*ko/N);
MP=real(MP); %The real output signal

% Plot output automatically for short inputs
if cc<=20
        t=0:length(nn)-1;
        t=t/fs;
        plot(t,real(MP))
end
end
```

APPENDIX II

```
 %function   [Mr,Y]=dmtdmd(type,q,KK,N,rb,del_f,y)
%
%                    DMT De-modulator
%                    Written by: Paul H. Moose
%                    Naval Postgraduate School
%                    Monterey, CA.
%
%      This m-file demodulates M-ary FSK or OFDM. It is implemented
% using with an fft  which is equivalent to a bank of digital correlators
% for each of the carriers.
%
%INPUTS:
%      type = 'mfsk' or 'ofdm' to specify the modulation
%      q =    no. of bits carrier in the M-ary FSK or no. of bits
%             carried by each of the OFDM carriers.
%      KK =   no. of carriers used in each OFDM symbol. Set to zero
%             for M-FSK.
%      N =    no. of points to be used in the fft.
%      rb =   symbol rate for M-FSK. Number of points in guard interval
%             for OFDM.
%      del_f = frequency separation of carriers
%      y =    matrix of input time domain symbols. Each column contains
%             complex baseband samples for one symbol.
%
%OUTPUTS:
%      Mr =   Matrix of recieved modulation values. Each column is a
%             vector containing the complex modulation values of the
%             M=2^q carriers for mfsk or the KK carriers for ofdm
%      Y =    Matrix of the fft of y after y has been zero padded to N
%             in the case of mfsk or after removing guard intervals
%             in the case of ofdm.
%
%USEAGE:
%      function   [Mr,Y]=dmtdmd(type,q,KK,N,rb,del_f,y)
%
%=====================================================================
========
function [Mr,Y]=dmtdmd(type,q,KK,N,rb,del_f,y)
%=====================================================================
```

```
=========
%Initialize
[rr,cc]=size(y);
%=====================================================================
=========
%=====================================================================
=========
%M-FSK: Extend symbols to length N and take fft.

if type=='mfsk'
        if rem(del_f,rb)==0
                P=del_f/rb;
                Ns=N;
        else
                cycles=del_f/rb;
                P=fix(cycles)+1;
                fraction=P-cycles;
                Ns=N-round(fraction*N/P);
        end

        fs=Ns*rb

        DEL_F=fs*P/N

        ye=[y;zeros(N-rr,cc)];
        Y=fft(ye);


%=====================================================================
=========
% Extract KL digital carriers from the  frequency domain array of N digital
% carrier frequencies and place in the received array R.
% Of the KL freqs in R, 2^q are the matched filter outputs of the mfsk signal.
P
        KL=P*(2^q-1)+1;
        K=floor(KL/2)
        R=ifreqa(K,Y);

%=====================================================================
=========
%Sample the filter outputs to obtain the 2^q modulation values for each
%symbol.
        [rr,cc]=size(R);
        Mr=R(1:P:2^q*P,:);
```

```
%=============================================================
========
%=============================================================
========
elseif type=='ofdm'

        %The precursor is removed from the input symbols and the fft is
        %computed

        y=y(rb+1:rb+N,:);

        Y=fft(y);
%=============================================================
==============

% The modulation values of the KK digital carriers are extracted and
% placed in the columns of array R.

        K=floor(KK/2);

        R=ifreqa(K,Y);
        R=R(1:KK,:);

%=============================================================
============
%Differentially decode the symbols in the time domain.
%       [rr,cc]=size(R);

                for l=1:cc-1
                        Mr(:,l)=R(:,l+1).*conj(R(:,l));
                end
%=============================================================
================



end
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd, STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library, Code 52 Naval Postgraduate School 411 Dyer Road Monterey, CA 93943-5101	2
3.	Research Office, Code 09 Naval Postgraduate School 589 Dyer Road Monterey, CA 93943-5138	1
4.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	1
5.	Dr. Paul H. Moose, Code EC/Me Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	4
6.	Dr. Herschel H. Loomis, Jr., Code EC/Lm Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	1
7.	Dr. Richard North, Code D855 NRad 53560 Hull Street San Diego, CA 92152	2